**TEST**

**R1**

**Subiectul 1**

Ce erori va genera codul urmator? Explicati de ce. Modificati/adaugati cod astfel incat aplicatia sa nu mai dea erori .

**using System;  
   
namespace Aplicatie  
{  
  class Test  
   {  
       
     private string name;**

**int age;**

**public  Test(string n,int a)**

**{**

**name=n;**

**age=a;**

**}    
     public void Print()   
      {  
        Console.WriteLine("\nMy name is " + name+ Age:" + age);  
      }**

**public void AcceptDetails()  
      {  
        Console.Write("\nEnter your age:");   
        age =  Console.ReadLine();   
      }**

**}  
   
  class Program  
   {  
     static void Main(string[] args)  
      {**

**Test ac, t;**

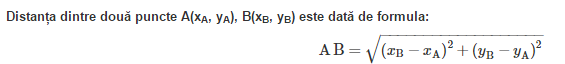
**ac = new Test();  
        Console.Write("Enter your name:\t");  
        ac.name = Console.ReadLine();  
        ac.Print();**

**t.AcceptDetails();   
        Console.ReadLine();  
      }  
   }  
}**

**Subiectul 2**

Scrieti o aplicatie care contine doua clase: **Punct** si **Test**.

Definiti o clasa **Punct** (punct in plan) care sa contina urmatoarele   
- doua campuri private x si y de tip double (care vor retine coordonatele punctului in plan)  
- proprietati de acces la cele 2 campuri x si y   
- constructor fara parametrii care sa-i initilalizeze pe x si y cu 0   
- constructor cu 2 parametrii double a si b care sa initializeze pe x cu a si pe y cu b   
- functie de citire de la tastatura a valorilor coordonatelor punctului  
- functie de afisare pe ecran a coordonatelor punctului intre paranteze si separate prin virgula  
- supraincarcarea operatorului - (minus) cu precizarea ca A-B este distanta de la punctul A la punctul B



Clasa **Test** va contine cod care creeaza obiecte de tip **Punct** si foloseste corect toate functiile definite.

**TEST**

**R2**

**Subiectul 1**

Ce erori va genera codul urmator? Explicati de ce. Modificati/adaugati cod astfel incat aplicatia sa nu mai dea erori .

**using System;  
   
namespace Static\_var\_and\_fun  
{  
  class Number  
   {  
     int num;**

**public  Number(int n)**

**{**

**num=n;**

**}**

**public static void Power()**

**{ Console.WriteLine("Power of {0} = {1}", num, num \* num);  
        Console.ReadLine();  
      }  
   }  
  class Program  
   {  
     static void Main(string[] args)  
      {**

**Number num=new Number();**

**Console.Write("Enter a number\t");  
        num = Console.ReadLine();   
        num.Power();   
      }  
   }  
}**

**Subiectul 2**

**Informatie**: **un număr complex în forma sa algebrică: z = a+i\*b, unde a este partea reală, iar b partea imaginară.**

Scrieti o aplicatie care contine doua clase: **Complex** si **Test**.

Definiti o clasa **Complex** care va conţine :

1. două câmpuri private de tip int : Re şi Im (reprezentând partea reală, respectiv partea imaginară a unui număr complex)
2. proprietati de acces la cele 2 campuri
3. un constructor fără parametrii care va iniţializa câmpurile la 0
4. un constructor cu doi parametrii întregi care va iniţializa partea reală cu valoarea primului parametru şi partea imaginară cu valoarea celui de-l doilea parametru
5. functie de citire de la tastatura a valorilor partilor reale si imaginare a numarului complex
6. functie de afisare pe ecran  a numarului complex (De exemplu: 3+i\*2. Atentie! Daca partea imaginară este 0 se va afisa doar partea reala (de ex. 3) şi dacă partea reală este 0 se va afisa de ex. doar 2\*i. Dacă partea imaginară a numărului complex este negativă, de ex. Re=2 şi Im=-3 atunci nu se va afişa 2+i\*(-3), va trebui să afişaţi 2-i\*3.)
7. o metodă care supraîncarcă operatorul + (suma a două numere complexe z1=a+i\*b şi z2=c+i\*d se calculează astfel: z1+z2=(a+i\*b)+(c+i\*d) = (a+c) + i\*(b+d) (deci partea reala Re va fie gala cu a+c, iar cea imaginara Im va fie gala cu b+d)

Clasa **Test** va contine cod care creeaza obiecte de tip **Complex** si foloseste corect toate functiile definite.